

WHAT IS CLAIMED IS:

- 1 1. A method of dynamically verifying program operation, comprising:
2 executing a specified computer program;
3 while executing the specified computer program, maintaining a shadow array, the
4 shadow array having entries corresponding to respective memory locations used by the
5 specified computer program, each entry of the shadow array indicating a data type of the
6 corresponding respective memory location;
7 the execution of the specified computer program including executing each of a
8 plurality of instructions of the computer program, wherein execution of each instruction of a
9 subset of the plurality of instructions includes:
10 determining whether execution of the instruction is inconsistent with an entry
11 of the shadow array and generating a report when execution of the instruction is determined to
12 be inconsistent with the entry of the shadow array;
13 executing the instruction; and
14 updating the shadow array in accordance with execution of the instruction.
- 1 2. The method of claim 1 wherein the execution of each instruction in the subset of the
2 plurality of instructions includes:
3 identifying a memory location to be accessed by the instruction;
4 inspecting the shadow array entry corresponding to the identified memory
5 location; and
6 determining whether execution of the instruction is inconsistent with the
7 inspected shadow array entry.
- 1 3. The method of claim 2 wherein access of the memory location by the instruction
2 comprises a read operation.
- 1 4. The method of claim 2 wherein access of the memory location by the instruction
2 comprises a write operation.

10040774-13301

1 5. The method of claim 1 wherein the determining includes determining whether proper
2 execution of the instruction requires accessing data of a predefined data type that is different
3 from the data type specified by the entry of the shadow array.

1 6. The method of claim 1 wherein the determining includes determining whether proper
2 execution of the instruction is inconsistent with the data type specified by the entry of the
3 shadow array.

1 7. The method of claim 1 wherein the respective memory locations include CPU
2 registers, stack locations, and memory heap locations.

1 8. The method of claim 1 wherein the data type indicated by at least a subset of the
2 shadow array entries indicates whether the corresponding memory location has been allocated.

1 9. The method of claim 1 wherein the data type indicated by at least a subset of the
2 shadow array entries indicates whether the corresponding memory location has been
3 initialized.

1 10. The method of claim 1, further comprising:
2 compiling a source code program into the specified computer program;
3 obtaining debugging information related to the specified computer program; and
4 initializing the shadow memory based on the debugging information.

1 11. The method of claim 1, further comprising not executing the instruction when
2 execution of the instruction is determined to be inconsistent with the entry of the shadow
3 array.

1 12. A computer program product for use in conjunction with a computer system, the
2 computer program product comprising a computer readable storage medium and a computer
3 program mechanism embedded therein, the computer program mechanism comprising:
4 a specified computer program;

5 a shadow array module for maintaining a shadow array, the shadow array having
6 entries corresponding to respective memory locations used by the specified computer
7 program, each entry of the shadow array indicating a data type of the corresponding respective
8 memory location; and

9 an interpreter module for executing the specified computer program including
10 executing each of a plurality of instructions of the specified computer program, wherein
11 execution of each instruction of a subset of the plurality of instructions includes:
12 determining whether execution of the instruction is inconsistent with an entry
13 of the shadow array and generating a report when execution of the instruction is determined to
14 be inconsistent with the entry of the shadow array;
15 executing the instruction; and
16 updating the shadow array in accordance with execution of the instruction.

1 13. The computer program product of claim 12 wherein the execution of each instruction
2 in the subset of the plurality of instructions includes:

3 identifying a memory location to be accessed by the instruction;
4 inspecting the shadow array entry corresponding to the identified memory
5 location; and
6 determining whether execution of the instruction is inconsistent with the
7 inspected shadow array entry.

1 14. The computer program product of claim 13 wherein access of the memory location by
2 the instruction comprises a read operation.

1 15. The computer program product of claim 13 wherein access of the memory location by
2 the instruction comprises a write operation.

1 16. The computer program product of claim 12 wherein the determining includes
2 determining whether proper execution of the instruction requires accessing data of a
3 predefined data type that is different from the data type specified by the entry of the shadow
4 array.

1 17. The computer program product of claim 12 wherein the determining includes
2 determining whether proper execution of the instruction is inconsistent with the data type
3 specified by the entry of the shadow array.

4 18. The computer program product of claim 12 wherein the respective memory locations
5 include CPU registers, stack locations, and memory heap locations.

1 19. The computer program product of claim 12 wherein the data type indicated by at least
2 a subset of the shadow array entries indicates whether the corresponding memory location has
3 been allocated.

1 20. The computer program product of claim 12 wherein the data type indicated by at least
2 a subset of the shadow array entries indicates whether the corresponding memory location has
3 been initialized.

1 21. The computer program product of claim 12, further comprising a compiling and
2 debugging module for compiling a source code program into the specified computer program,
3 and wherein the shadow array module further:
4 obtains debugging information related to the specified computer program from the
5 compiling and debugging module; and
6 initializes the shadow memory based on the debugging information.

1 22. The computer program product of claim 12, further comprising not executing the
2 instruction when execution of the instruction is determined to be inconsistent with the entry of
3 the shadow array.

1 23. A computer program product for use in conjunction with a computer system, the
2 computer program product comprising a computer readable storage medium and a computer
3 program mechanism embedded therein, the computer program mechanism comprising:
4 a program instrumenting module for adding dynamic checking instructions to a
5 compiled program to generate an instrumented program, the dynamic checking instructions
6 including instructions for establishing a shadow array, the shadow array having entries

7 corresponding to respective memory locations used by the compiled program, each entry of
8 the shadow array indicating a data type of the corresponding respective memory location;
9 the compiled program including a plurality of instructions;
10 wherein the dynamic checking instructions are configured so that during execution of
11 instructions of the instrumented program, for each instruction of a subset of the plurality of
12 instructions of the compiled program, the dynamic checking instructions:
13 determine whether execution of the instruction of the compiled program is
14 inconsistent with an entry of the shadow array and generate a report when execution of the
15 instruction is determined to be inconsistent with the entry of the shadow array; and
16 update the shadow array in accordance with execution of the instruction of the
17 compiled program.

1 24. The computer program product of claim 23 wherein the determining comprises:
2 identifying a memory location to be accessed by the instruction;
3 inspecting the shadow array entry corresponding to the identified memory
4 location; and
5 determining whether execution of the instruction is inconsistent with the
6 inspected shadow array entry.

1 25. The computer program product of claim 24 wherein access of the memory location by
2 the instruction comprises a read operation.

1 26. The computer program product of claim 24 wherein access of the memory location by
2 the instruction comprises a write operation.

1 27. The computer program product of claim 23 wherein the determining includes
2 determining whether proper execution of the instruction requires accessing data of a
3 predefined data type that is different from the data type specified by the entry of the shadow
4 array.

1 28. The computer program product of claim 23 wherein the determining includes
2 determining whether proper execution of the instruction is inconsistent with the data type
3 specified by the entry of the shadow array.

1 29. The computer program product of claim 23 wherein the respective memory locations
2 include CPU registers, stack locations, and memory heap locations.

1 30. The computer program product of claim 23 wherein the data type indicated by at least
2 a subset of the shadow array entries indicates whether the corresponding memory location has
3 been allocated.

1 31. The computer program product of claim 23 wherein the data type indicated by at least
2 a subset of the shadow array entries indicates whether the corresponding memory location has
3 been initialized.

1 32. The computer program product of claim 23, further comprising a compiling and
2 debugging module for compiling a source code program into the compiled program, and
3 wherein the dynamic checking instructions further:
4 obtain debugging information related to the compiled program from the compiling and
5 debugging module, and
6 initialize the shadow memory based on the debugging information.

1 33. The computer program product of claim 23, further comprising not executing the
2 instruction when execution of the instruction is determined to be inconsistent with the entry of
3 the shadow array.